DOCKET NO.: MSFT-1166/191807.1 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Angus Dorbie** Confirmation No.: **8083**

Serial No.: **09/247,816** Group Art Unit: **2671**

Filing Date: **February 9, 1999** Examiner: **Huedung Cao**

For: **METHOD AND APPARATUS FOR EARLY CULLING OF OCCLUDED OBJECTS**

EXPRESS MAIL LABEL NO: EL969195249US
DATE OF DEPOSIT: September 12, 2003

Mail Stop Appeal-Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPELLANT'S BRIEF PURSUANT TO 37 C.F.R. § 1.192

This brief is being filed in support of Appellant's appeal from the rejections of claims 1-18 dated October 21, 2002. A Notice of Appeal was filed on April 21, 2003.

1. REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation by virtue of an assignment from Silicon Graphics, Inc. to Microsoft Corporation, which was filed on December 2, 2002. Angus Dorbie (Appellant) initially assigned the application to Silicon Graphics Inc., the assignment being recorded on February 9, at Reel 009784, Frame 0600.

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

3. STATUS OF CLAIMS

Claims 1-18 are pending in the present application. Claims 1, 4, 6, 11-13 stand rejected under 35 U.S.C. §103 (a) as allegedly being obvious over U.S. Patent 5,977,980 (*Aleksicy*) in view of U.S. Patent 5,596,686 (*Duluk, Jr.*). Claims 14-16 and 18 stand rejected under 35 U.S.C. §103(a) as allegedly being obvious over U.S. Patent 5,579,455 (*Green et al.*). Claims 2-3, 5, 7-10, stand rejected under 35 U.S.C. §103(a) as allegedly being obvious over *Aleksicy* in view of *Duluk, Jr.* and further in view of U.S. Patent 6,091,422 (*Ouaknine et al.*).

Claims 1-18 are reproduced in Appendix A, attached hereto, with an indication of status next to each claim, as of the date of this appeal.

4. STATUS OF AMENDMENTS

No Amendments to claims 1-18 have been filed subsequent to the final rejection.

5. SUMMARY OF INVENTION

Prior to Appellant's invention, the task of rendering primitives of a graphics pipeline into graphical display pixels was extremely time consuming. This is

especially true for complex images that include many primitives or require complex lighting, shading or other effects. Such time consumption transforming primitives becomes problematic for applications, such as flight simulators and virtual reality environments, where rapid image generation is required.

Advantageously, with Appellant's invention, the transformation of primitives to pixels is optimized through the use of early occlusion culling methods. Generally, occlusion culling methods attempt to improve the speed of the rendering process. For methods of this type, the rendering process is modified to take advantage of the fact that images often contain overlapping objects. The result is an image having objects that are fully or partially hidden or occluded. In operation, occlusion culling methods detect occluded objects and eliminate them from all or part of the rendering process.

In operation, the present invention divides a display screen into a series of tiles arranged as a rectangular grid. The rectangular grid is known as a coarse Z-buffer and may have various sizes and dimensions. Each tile within the coarse Z-buffer has an associated depth value. Each tile's depth value is defined as the farther Z-buffer value that is included within that tile.

When processing primitives, the graphics pipeline is configured to update the depth values using information fed back from the Z-buffer. To maximize the effectiveness of the occlusion culling method, the graphics pipeline may be configured to perform these updates on a synchronous basis. Hence, the depth values are updated each time the corresponding Z-values in the Z-buffer are changed. Additionally, the graphics pipeline may also be configured to perform these updates on a less frequent basis. Asynchronous updating provides a balanced approach that retains most of the benefit of the occlusion culling method while reducing the amount of data that must be fed back from the Z-buffer.

The depth values are stored in a memory location (e.g. main memory) where they are available to application programs. This allows the application programs to reference these values while they are creating graphics images. The program rendering an image constructs a surrogate volume for each object that it adds to the image. The program then

compares the nearest Z-value of the surrogate volume to the depth value of the tile that includes the surrogate volume. Based on this comparison, the application program determines if the object is occluded and can be discarded.

In some cases, a surrogate volume may span several tiles. In these cases the application program may determine if the object is occluded by performing comparisons with depth values for each of the spanning tiles. Alternately, it is possible to provide a series of coarse Z-buffers, each containing depth values for a different resolution. In such an arrangement, even large surrogate volumes may be analyzed using a single comparison.

The testing of surrogate volumes and the discarding of occluded objects is performed by the application program, executing on a host processor. Furthermore, the application program performs this test for each object before the object is sent to the graphics processor. As a result, objects that are found to be occluded never reach the graphics processor. The graphics processor is freed from any processing associated with these object.

6. ISSUES

I. Whether claims 1, 4, 6, 11-13 patentably define over *Aleksicy* in view of *Duluk, Jr.*?

II. Whether claims 14-16 and 18 patentably define over *Greene et al.*?

III. Whether claims 2-3, 5, 7-10 patentably define over *Aleksicy* in view of *Duluk, Jr.* and further in view *Ouaknine et al.*?

7. GROUPING OF CLAIMS

Group I: Claims 1-10 stand or fall together. Each of these claims calls for the features of "culling occluded objects from an image being rendered into a frame buffer by constructing a coarse Z-buffer, the coarse Z-buffer being subdivided into a series of tiles, each tile having an associated depth value; updating the depth values of the coarse Z-

buffer using Z information from the frame buffer; and using the depth values to selectively discard the occluded objects from the image being rendered."

Group II: Claim 11 stands alone. Claim 11 calls for the features of "a host processor executing a graphics application program, wherein the graphics application program is capable to implement: a generation stage for creation, acquisition, and modification of information to be displayed, and organizing the information into application data structures; and a traversal stage for traversal of the application data structures, and passing on appropriate graphics data; and a graphics processor, communicatively couples to the host processor, capable to implement: a transformation stage for transformation of graphics data from object space coordinates into eye-space coordinates, performing requested lighting operation, clipping the transformed data in clip-space, and projecting resulting coordinates into window-space; a rasterization stage for rendering window-space primitives into a frame buffer, and performing shading calculations, texture lookups and calculations, and per-pixel operations; a feedback loop permitting the rasterization stage to return information to the traversal stage; and a display stage for scanning resulting pixels in frame buffer for display to a display device."

Group III: Claims 12 and13 stand or fall together. Each of these claims calls for the features of "means for constructing a coarse Z-buffer, the coarse Z-buffer subdivided into a series of tiles, each tile having an associated depth value; means for updating the depth values of the coarse Z-buffer using Z information from a frame buffer; and means for using the depth values to selectively discard occluded objects from an image being rendered into the frame buffer."

Group IV: Claims 14-17 stand or fall together. Each of these claims calls for the features of "ordering all objects, the objects being included in an image being rendered, according to their distance from eye point by i) logically dividing area of the image into a

coarse Z-buffer, the coarse Z-buffer including a series of tiles, the tiles being arranges in an rectangular grid, wherein the grid may have different resolutions, and wherein each tile has an associated depth value, the depth value being a Z-buffer value farthest from the eye that is included within that tile; ii) constructing a surrogate volume for each object of the image, wherein each surrogate volume is a three-dimensional object that is just large enough to contain the object being ordered and wherein each surrogate volume may span only one tile of tan appropriate resolution; iii) determining a depth value of the surrogate volume that is nearest the eye of the viewer; iv) determining the depth value of the one tile that includes the surrogate volume; and v) comparing the depth value of the surrogate volume versus the depth value of the tile including the surrogate volume;

culling the object whose surrogate volume has a depth value farther from the eye than the depth value of the tile, including the surrogate volume, after a single comparison; and

rendering the objects whose surrogate volume has a depth value closer to the eye than the depth value of the tile, including the surrogate volume, or equidistant to the eye with the depth value of the tile including the surrogate volume."

Group V: Claim 18 stands alone. Claim 18 calls for the features of "a memory for storing depth values, wherein the depth values are derived by: a) ordering all objects, the objects being included in an image being rendered, according to their distance from eye point, accomplished by, i) logically dividing area of the image into a coarse Z-buffer, the coarse Z-buffer including a series of tiles, the tiles being arranged in a rectangular grid, wherein the grid may have different resolutions, and wherein each tile has an associated depth value, the depth value being a Z-buffer value farthest from the eye that is included within that tile; ii) constructing a surrogate volume for each object of the image, wherein each surrogate volume is a three-dimensional object that is just large enough to contain the object being ordered and wherein each surrogate volume may span only one tile of an appropriate resolution; iii) determining a depth value of the surrogate

volume that is nearest to eye of a viewer; iv) determining a depth value of the one tile that includes the surrogate volume;

a host processor for:

determining which objects are occluded by comparing the depth value of the surrogate volume versus the depth value of the tile including the surrogate volume,

generating the surrogate volumes for the objects being processed, and transforming the surrogate volumes from object space to eye space; and a graphics processor for rendering objects that are not occluded."

8. ARGUMENT

In the Office Action, dated October 21, 2002, the Examiner finally rejected claims 1-18 as being unpatentable over various combinations of the *Aleksicy*, *Duluk, Jr.*, *Ouaknine*, and *Green et al.* In the present section, Appellant presents arguments as to why, even if *Aleksicy*, *Duluk, Jr.*, *Ouaknine*, and *Green et al.* were properly combinable, neither *Aleksicy*, *Duluk, Jr.*, *Ouaknine*, nor *Green et al.*, taken alone or in combination, teach or suggest the recited features of Appellant's invention, which provides for the early culling of occlusion in a graphics pipeline.

Differences from *Aleksicy*

Aleksicy discloses an apparatus and method for determining visibility of a pixel during video rendering. The visibility is accomplished by determining Z-positioning information of an object element that is being rendered. The Z-positioning information is representative of the Z-information of an object element in a particular section of the display. For example. The Z-positioning information may be the two most significant bits of the Z-parameter, or information of an object element.

In contrast to the inventions of Appellant's claim 1, however, the systems and methods taught by *Aleksicy* does not teach a system and methods that provide early occlusion culling wherein a display area is modeled as a coarse Z-buffer being having a make-up of tiles, wherein each tile is <u>associated a depth value, and wherein the depth</u>

value is used by applications to compare objects to be rendered so as to remove occluded objects. Rather, *Aleksicy* teaches an apparatus and method that assigns z-positioning information to object elements to be rendered in a display area. Furthermore, as Examiner states *Aleksicy* "**does not explicitly disclose the coarse Z-buffer subdivide into a series of tiles.**" (Page 2 of Official Action dated January 29, 2002 – Paper 8).

In the Official Final Rejection Action, it is then stated that *Aleksicy* teaches the use of a tiled coarse Z-buffer. (See Official Final Rejection Action – Page 7, "Response to Arguments" section). Appellant respectfully submits that the Official Final Rejection Action is in direct contradiction with the statements made previously on the record. Appellant respectfully submits that such contradiction is an effort to sanitize the record in response to Appellant's responses. Appellant respectfully submit that, *Aleksicy* does not teach the use of a **coarse tiled z-buffer** when processing graphics information to realize occluded culling.

Differences from *Duluk*

Duluk discloses an apparatus and method for a disclose a system and method for a parallel query Z-coordinate buffer. The apparatus and method perform a keep/discard decision on screen coordinate geometry before the geometry is converted or rendered into individual display screen pixels by implementing a parallel searching technique within a novel z-coordinate buffer based on a magnitude comparison content addressable memory (MCAMM) structure. In operation, the MCAMM provides a means for performing simultaneous arithmetic magnitude comparisons on numerical quantities.

In contrast to the inventions of Appellant's claim 1,11, 12, 13, 14, and 18, however, the systems and methods taught by *Duluk* does not teach a system and methods that provide early occlusion culling wherein a display area is modeled as a coarse z-buffer being having a make-up of tiles, wherein each tile is associated a depth value, and wherein the depth value is used by applications to compare objects to be rendered so as to remove occluded objects. Rather, *Duluk* teaches an apparatus and method that perform comparisons on *already known* values for content queried from a z-buffer in a

determination of keeping or discarding graphical content before rendering. ***Duluk does not explicitly or implicitly teach assigning depth values of a tile processed graphical display area as part of an effort of determining and discarding occluded graphical content.***

Differences from *Ouaknine et al.*

Ouaknine et al. disclose a computer 3D modeling and animation system that provides a user interface that allows users to make changes to 3D scene data while simultaneously allowing the author to view rendered images corresponding to those changes. The system provides tools associated with a 3D editing context as well as a rendered image that is automatically updated. In operation, the speed of updating the rendering is increased by identifying and re-rendering only portions of the region that require it. This may be achieved by breaking an image to be rendered into tiles, some or portions of which may be subtended by the render region. The invention , in this case, may determine from the dimensions (coordinates) of the tiles, the particular scene modifications that are still current, and from the other scene data, the particular tiles that must be re-rendered.

In contrast to the inventions of Appellant's claim 1,11, 12, 13, 14, and 18, however, the systems and methods taught by *Ouaknine et al.* do not teach a system and methods that provide early occlusion culling wherein a display area is modeled as a coarse z-buffer being having a make-up of tiles, wherein each tile is associated a depth value, and wherein the depth value is used by applications to compare objects to be rendered so as to remove occluded objects. Rather, *Ouaknine et al.* teach a system that allows for the more efficient editing of 3D images wherein images are broken down into tiles. The tiles are then processed during rendering to determine which portions of the image require modification (as a result of editing) and those that do not. *Ouaknine et al.*, however, ***does not explicitly or implicitly teach assigning depth values of a tile processed graphical display area as part of an effort of determining and discarding occluded graphical content.***

<u>Differences from *Greene et al.*</u>

Greene et al. disclose a hierarchical Z-buffer scan-conversion algorithm that rejects hidden geometry of a model, and exploits the spatial and temporal coherence of the images being generated. The method employs two hierarchical data structures, an object-space octree and an image-space Z-pyramid, in order to accelerate scan conversion. The two hierarchical data structures make it possible to reject hidden geometry while rendering visible geometry with the speed of scan conversion. In operation, an image-space Z-pyramid id employed. The z-pyramid offers the ability to make a determination whether a large polygon is hidden, making it unnecessary to scan-convert the polygon. This may be accomplished by combining four Z values (a 2x2) window at each level into one Z value at the next coarser level by choosing the farthest Z from the observer.

In contrast to the inventions of Appellant's claims 1,11, 12, 13, 14, and 18 however, the systems and methods taught by *Greene et al.* do not teach a system and methods that provide early occlusion culling wherein a display area is modeled as a coarse z-buffer being having a make-up of tiles, wherein each tile is <u>associated a depth value, and wherein the depth value is used by applications to compare objects to be rendered so as to remove occluded objects</u>. Rather, *Greene et al.* teach an algorithm that processes graphical content to remove hidden elements of an object model and to render visible geometry at the speed of scan conversion. *Greene et al.*, however, do not explicitly or implicitly teach assigning depth values of a tile processed graphical display area as part of an effort of determining and discarding occluded graphical content.

<u>Claim Analysis (Groups I – V):</u>

<u>Independent Claims 1 and 11-13:</u>

Independent claims 1, 11-13 stand rejected as allegedly being obvious over *Aleksicy* and *Duluk*. Specifically, in the reading of the *Aleksicy*, the record suggests that *Aleksicy* teaches a method for culling occluded objects from an image being rendered into a frame buffer, the method being performed by a host processor and comprising the steps

of constructing a coarse Z-buffer, the coarse Z-buffer being subdivided into a series of tiles each having an associated depth value. (Page 2 of Current Action citing to col. 2, lines 25-31, and Figure 1 of *Aleksicy*). Furthermore, the record suggests that *Duluk* teaches tile processing of graphical content. (Page 2 of Current Action).

Appellant respectfully disagrees with the interpretation of the *Aleksicy* and *Duluk* references. Specifically, *Aleksicy* teaches **the determination of visibility of a pixel during rendering by determining the z-positioning information the on object element.** Appellant respectfully submits that one of ordinary skill in the art would not equate the use of z-positioning information to determine pixel visibility to the tile processing of graphical content and association of depth information for tile processed graphical content to determine occluded content. (See Claims 1 and 11-14, and 18). It is apparent that these steps operate very differently and thus are not similar or the same.

Comparatively, *Duluk* teaches a system and method that perform a keep/discard decision on screen coordinate geometry using a z-coordinate buffer based on MCCAM structure. However, as described previously, *Duluk*, fails to teach the association of depth values to tile processed graphical content when realizing the keep/discard decision.

Since the use of Z-positioning information does not constitute tile processing of graphical content and association of depth information to tile-processed information in determining occluded content, Appellant respectfully submits that *Aleksicy* and *Duluk*, alone or in combination, do not render the independent claims 1, and 11-13 obvious.

Independent Claims 14 and 18:

Independent claims 14, and 18 stand rejected as allegedly being obvious over *Greene et al.* Specifically, the records suggest that *Greene et al.* teaches all of the limitations of independent claims 14 and 18 of the present application.

Appellant respectfully disagrees with how the *Greene et al.* reference has been interpreted on record. *Greene et al.* teaches,

> A hierarchical Z-buffer scan-conversion algorithm that does well on both (a) quickly rejecting most of the hidden geometry in a model, and (b) exploiting the

> spatial and temporal coherence of the images being generated. The method uses two hierarchical data structures, an object-space octree and an image-space Z-pyramid, in order to accelerate scan conversion. The two hierarchical data structures make it possible to reject hidden geometry very rapidly while rendering visible geometry with the speed of scan conversion. For animation purposes, the algorithm is also able to exploit temporal coherence. The resulting method is well suited to models with high depth complexity, achieving significant speedup in some cases compared to ordinary scan conversion. (See *Greene et al.* ABSTRACT).

Appellant respectfully submits that independent claims 14 and 18 teach the ordering all objects, the objects being included in an image being rendered, according to their distance from eye point, by (i) logically dividing area of the image into a coarse Z-buffer, the coarse Z-buffer including a series of tiles, the tiles being arranges in an rectangular grid, wherein the grid may have different resolutions, and wherein each tile has an associated depth value, the depth value being a Z-buffer value farthest from the eye that is included within that tile; (ii) constructing a surrogate volume for each object of the image, wherein each surrogate volume is a three-dimensional object that is just large enough to contain the object being ordered and wherein each surrogate volume may span only one tile of tan appropriate resolution; (iii) determining a depth value of the surrogate volume that is nearest the eye of the viewer; (iv) determining the depth value of the one tile that includes the surrogate volume; (iv) comparing the depth value of the surrogate volume versus the depth value of the tile including the surrogate volume such that the culling of the object whose surrogate volume has a depth value farther from the eye than the depth value of the tile, including the surrogate volume, after a single comparison is performed and the rendering of the objects whose surrogate volume has a depth value closer to the eye than the depth value of the tile, including the surrogate volume, or equidistant to the eye with the depth value of the tile including the surrogate volume is undertaken.

Nowhere does *Greene et al.* teach the very basic feature of ordering of objects according to their distance from the eye point. The record indicates that such teaching may be found at column 12, lines 51-59 of the *Greene* reference. However, Appellant

does not read this cite to teach what the record purports. Specifically, Col. 12, Lines 51-59 of the *Greene et al.* reference states:

> FIG. 6 is an overall flowchart of a procedure which may be used to implement the present invention. This embodiment incorporates all three primary aspects of the invention, and at various points in the description it is pointed out how the embodiment can be modified if it is desired to use less than all three primary aspects of the invention. The first few flowcharts illustrate how the invention can take advantage of temporal coherence in rendering sequential frames of a model. As used herein, a "frame" is the two-dimensional image which a viewer sees when viewing all of the surface primitives of a model from a particular viewpoint.

Simply, *Greene et al.* fails to teach all of the limitations of independent claims 14 and 18 (e.g. ordering of objects according to the distance from an eye point) and as such fails to render the present invention obvious.

Dependent Claims 2-10, and 15-17:

The record indicates that dependent claims 2-10, and 15-17 stand rejected as being allegedly obvious over *Aleksicy, Duluk, Ouakine et al.,* and *Greene et al.* for reasons set forth in the official record. Inasmuch as claims 2-10 and 15-17 depend either directly or indirectly from independent claims 1, 11-13, 14, and 18 , Appellant respectfully submits that they too patentably define over the prior art of record for the same reasons.

Conclusion

In sum, even if *Aleksicy, Duluk, Ouakine et al.,* and *Greene et al.* were combined, nowhere do *Aleksicy, Duluk, Ouakine et al.,* and *Greene et al.* disclose the bulk of Appellant's invention. Accordingly, reversal of the rejection to claim Groups I to V under 35 U.S.C. § 103 is respectfully requested.

For all the foregoing reasons, Appellant respectfully requests that the Board reverse the rejection of claims 1-18.

Respectfully submitted,

Date: September 12, 2003

George J. Awad
Registration No. 46,528

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

APPENDIX A

Claims on Appeal

1. (Previously Amended) A method for culling occluded objects from an image being rendered into a frame buffer, the method performed by the host processor comprising:

constructing a coarse Z-buffer, the coarse Z-buffer subdivided into a series of tiles, each tile having an associated depth value;

updating the depth values of the coarse Z-buffer using Z information from the frame buffer; and

using the depth values to selectively discard the occluded objects from the image being rendered.

2. (Previously Amended) A method as recited in claim 1, wherein updating depth values is performed synchronously as information in the frame buffer changes.

3. (Previously Amended) A method as recited in claim 1, wherein updating the depth values is performed asynchronously.

4. (Previously Amended) A method as recited in claim 1, wherein using the depth values to selectively discard objects further comprises:

constructing a surrogate volume for an object; and

comparing nearest Z-value of the surrogate volume to the depth value of a tile that includes the surrogate volume.

5. (Previously Amended) A method as recited in claim 4, further comprising transforming the surrogate volume from object space to eye space.

6. (Previously Amended) A method as recited in claim 1, using the depth values to selectively discard the occluded objects further comprises:

constructing a surrogate volume for an object; and

retrieving greatest depth value from the depth values set of tiles that are spanned by the surrogate volume; and

comparing the nearest Z-value of the surrogate volume to the retrieved depth value.

7. (Previously Amended) A method as recited in claim 6, further comprising the transforming the surrogate volume from object space to eye space.

8. (Previously Amended) A method as recited in claim 1, further comprising:

constructing a lower resolution coarse Z-buffer, the lower resolution coarse Z-buffer subdivided into a series of tiles, each tile having an associated depth value; and

updating the depth values of the lower resolution coarse Z-buffer using Z information form the frame buffer.

9. (Previously Amended) A method as recited in claim 8, wherein each tile in the lower resolution coarse Z-buffer covers the same screen area as each tile in the coarse Z-buffer.

10. (Previously Amended) A method as recited in claim 8, wherein the tiles in the lower resolution Z-buffer are overlapping.

11. (Previously Added) A system, used as a host for a graphics pipeline, comprising:

a host processor executing a graphics application program, wherein the graphics application program is capable to implement:

a generation stage for creation, acquisition, and modification of information to be displayed, and organizing the information into application data structures; and

a traversal stage for traversal of the application data structures, and passing on appropriate graphics data; and

a graphics processor, communicatively couples to the host processor, capable to implement:

a transformation stage for transformation of graphics data from object space coordinates into eye-space coordinates, performing requested lighting operation, clipping the transformed data in clip-space, and projecting resulting coordinates into window-space;

a rasterization stage for rendering window-space primitives into a frame buffer, and performing shading calculations, texture lookups and calculations, and per-pixel operations;

a feedback loop permitting the rasterization stage to return information to the traversal stage; and

a display stage for scanning resulting pixels in frame buffer for display to a display device.

12. (Previously Added) A system comprising:

means for constructing a coarse Z-buffer, the coarse Z-buffer subdivided into a series of tiles, each tile having an associated depth value;

means for updating the depth values of the coarse Z-buffer using Z information from a frame buffer; and

means for using the depth values to selectively discard occluded objects from an image being rendered into the frame buffer.

13. (Previously Added) A machine-readable medium comprising instructions to a machine to:

construct a coarse Z-buffer, the coarse Z-buffer subdivided into a series of tiles, each tile having an associated depth value;

update the depth values of the coarse Z-buffer using Z information from a frame buffer; and

use the depth values to selectively discard objects from an image being rendered into the frame buffer.

14. (Previously Added) A method for early culling of occluded objects, comprising:

a) ordering all objects, the objects being included in an image being rendered, according to their distance from eye point, comprising:

i) logically dividing area of the image into a coarse Z-buffer, the coarse Z-buffer including a series of tiles, the tiles being arranges in an rectangular grid, wherein the grid may have different resolutions, and wherein each tile has an associated depth value, the depth value being a Z-buffer value farthest from the eye that is included within that tile;

ii) constructing a surrogate volume for each object of the image, wherein each surrogate volume is a three-dimensional object that is just large enough to contain the object being ordered and wherein each surrogate volume may span only one tile of tan appropriate resolution;

iii) determining a depth value of the surrogate volume that is nearest the eye of the viewer;

iv) determining the depth value of the one tile that includes the surrogate volume;

v) comparing the depth value of the surrogate volume versus the depth value of the tile including the surrogate volume;

b) culling the object whose surrogate volume has a depth value farther from the eye than the depth value of the tile, including the surrogate volume, after a single comparison; and

c) rendering the objects whose surrogate volume has a depth value closer to the eye than the depth value of the tile, including the surrogate volume, or equidistant to the eye with the depth value of the tile including the surrogate volume.

15. (Previously Added) The method of claim 14, wherein the surrogate volume may span several tiles and further comprising:

comparing the depth value of the surrogate volume with each of the spanning tiles; and

culling the objects whose surrogate volume has a depth value farther from the eye than the depth value of the tiles including the surrogate volume; and

rendering the objects whose surrogate volume has a depth value closer to the eye than the depth value of at least one of the tiles including the surrogate volume or is equidistant to the eye with at least one of the tiles including the surrogate volume.

16. (Previously Added) The method of claim 14, further comprising:

subdividing the objects that are not occluded into smaller objects; and

determining if the smaller objects are occluded.

17. (Previously Added) The method of claim 14, wherein

each coarse Z-buffer is replicated one or more times at different resolutions, each separate coarse Z-buffer spans the image using a different resolution, the number of tiles in the coarse Z-buffers of various resolutions remains constant, for lower resolution coarse Z-buffers, each tile covers a larger area of the image, for lower resolution coarse Z-buffers, the tiles overlap one another,

center points of successive resolutions of tiles of the coarse Z-buffers are offset from the center points of preceding resolution of tiles,

lower resolution of the coarse Z-buffers split the image between tiles with overlap, a higher resolution coarse Z-buffer splits the image between tiles with no overlap, and a host processor is allowed to select a resolution that corresponds to a size of any given object.

18. (Previously Added) A system comprising:

a memory for storing depth values, wherein the depth values are derived by:

a) ordering all objects, the objects being included in an image being rendered, according to their distance from eye point, comprising:

i) logically dividing area of the image into a coarse Z-buffer, the coarse Z-buffer including a series of tiles, the tiles being arranged in a rectangular grid, wherein the grid may have different resolutions, and wherein each tile has an associated depth value, the depth value being a Z-buffer value farthest from the eye that is included within that tile;

ii) constructing a surrogate volume for each object of the image, wherein each surrogate volume is a three-dimensional object that is just large enough to contain the object being ordered and wherein each surrogate volume may span only one tile of an appropriate resolution;

iii) determining a depth value of the surrogate volume that is nearest to eye of a viewer;

iv) determining a depth value of the one tile that includes the surrogate volume;

a host processor for:

determining which objects are occluded by comparing the depth value of the surrogate volume versus the depth value of the tile including the surrogate volume,

generating the surrogate volumes for the objects being processed, and transforming the surrogate volumes from object space to eye space; and a graphics processor for rendering objects that are not occluded.

09-15-03 AF 2671

DOCKET NO.: MSFT-1166/191807.1 PATENT #20

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:
Angus Dorbie

Confirmation No.: 8083

Application No.: 09/247,816

Group Art Unit: 2671

Filing Date: February 9, 1999

Examiner: Huedung Cao

For: **Method And Apparatus For Early Culling Of Occluded Objects**

EXPRESS MAIL LABEL NO: EL 969195249US
DATE OF DEPOSIT: September 12, 2003

EL969195249US

MS Appeal Brief - Patent
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF TRANSMITTAL
PURSUANT TO 37 CFR § 1.192

Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal received by The United States Patent and Trademark Office on **April 21, 2003**.

☐ Applicant(s) has previously claimed small entity status under 37 CFR § 1.27 .

☐ Applicant(s) by its/their undersigned attorney, claims small entity status under 37 CFR § 1.27 as:

- ☐ an Independent Inventor
- ☐ a Small Business Concern
- ☐ a Nonprofit Organization.

☒ Petition is hereby made under 37 CFR § 1.136(a) (fees: 37 CFR § 1.17(a)(1)-(4) to extend the time for response from the Notice of Appeal which was filed on April 21, 2003 and which was received by the United States Patent and Trademark Office on

April 21, 2003 to and through **September 21, 2003** comprising an extension of the shortened statutory period of 3 months.

	SMALL ENTITY		NOT SMALL ENTITY	
	RATE	FEE	RATE	FEE
☒ APPEAL BRIEF	$160	$	$320	$320.00
☐ ONE MONTH EXTENSION OF TIME	$55	$	$110	$
☐ TWO MONTH EXTENSION OF TIME	$205	$	$410	$
☒ THREE MONTH EXTENSION OF TIME	$465	$	$930	$930.00
☐ FOUR MONTH EXTENSION OF TIME	$725	$	$1450	$
☐ FIVE MONTH EXTENSION OF TIME	$985	$	$1970	$
☐ LESS ANY EXTENSION FEE ALREADY PAID	minus	($)	minus	($)
TOTAL FEE DUE		$0		$1,250.00

☒ The Commissioner is hereby requested to grant an extension of time for the appropriate length of time, should one be necessary, in connection with this filing or any future filing submitted to the U.S. Patent and Trademark Office in the above-identified application during the pendency of this application. The Commissioner is further authorized to charge any fees related to any such extension of time to Deposit Account 23-3050. This sheet is provided in duplicate.

☒ A check in the amount of **$1,250.00** is attached. Please charge any deficiency or credit any overpayment to Deposit Account No. 23-3050.

☐ Please charge Deposit Account No. 23-3050 in the amount of **$ .00**. This sheet is attached in duplicate.

☒ The Commissioner is hereby requested to grant an extension of time for the appropriate length of time, should one be necessary, in connection with this filing or any future filing submitted to the U.S. Patent and Trademark Office in the above-identified application during the pendency of this application. The Commissioner is further authorized to charge any fees related to any such extension of time to deposit account 23-3050. This sheet is provided in duplicate.

Date: September 12, 2003

George J. Awad
Registration No. 46,528

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439